

Programmation NoSQL



Bienvenue!

Vous n'avez pas peur?

Nous allons programmer notre première base de données NoSQL

(ou presque, car les services web, et le localStorage en étaient)



NO SQL

Des bases de données non-relationnelles
ou qu'on accède sans le SQL

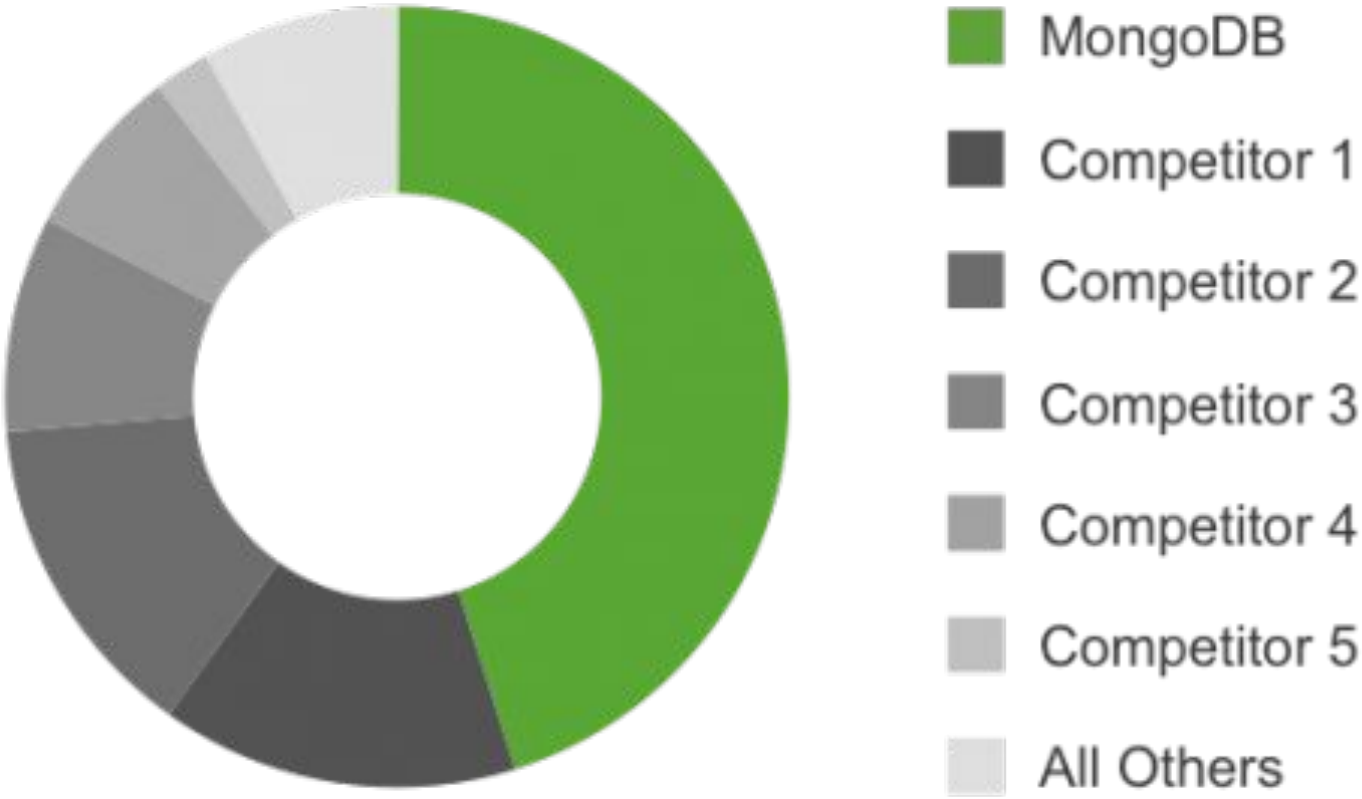
On teste à chaque étape du processus



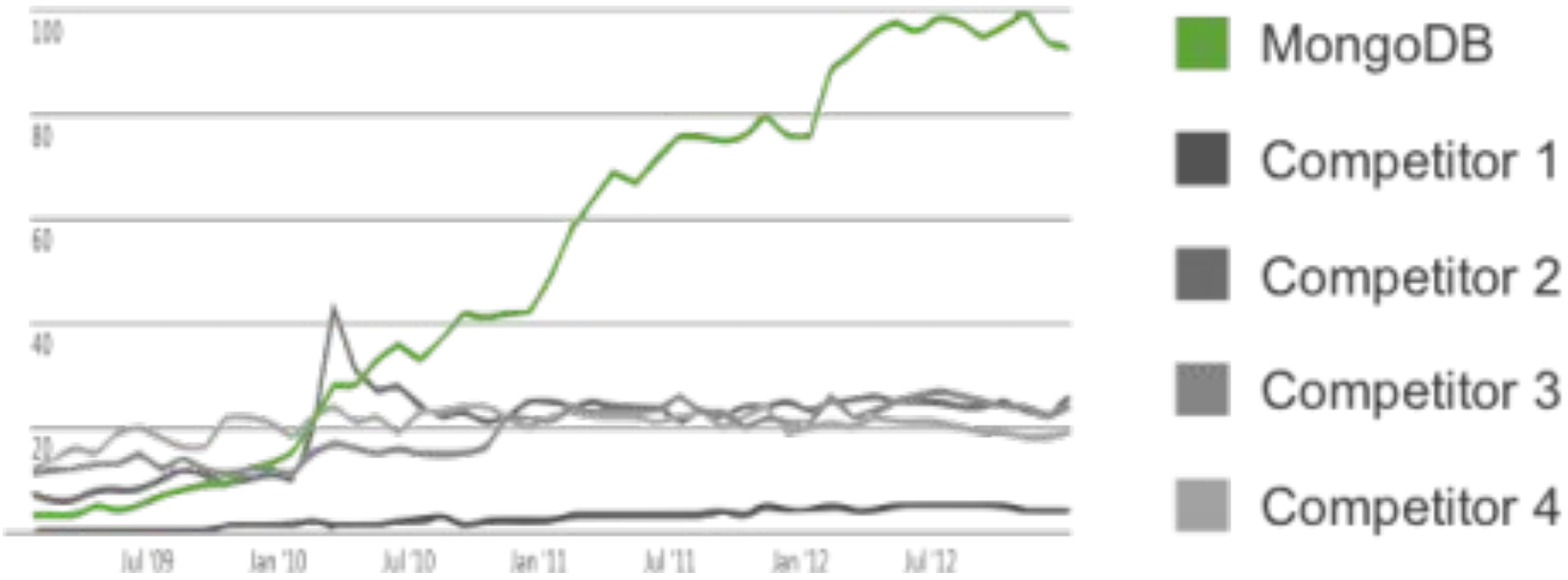
1.
MongoDB

Avec Java

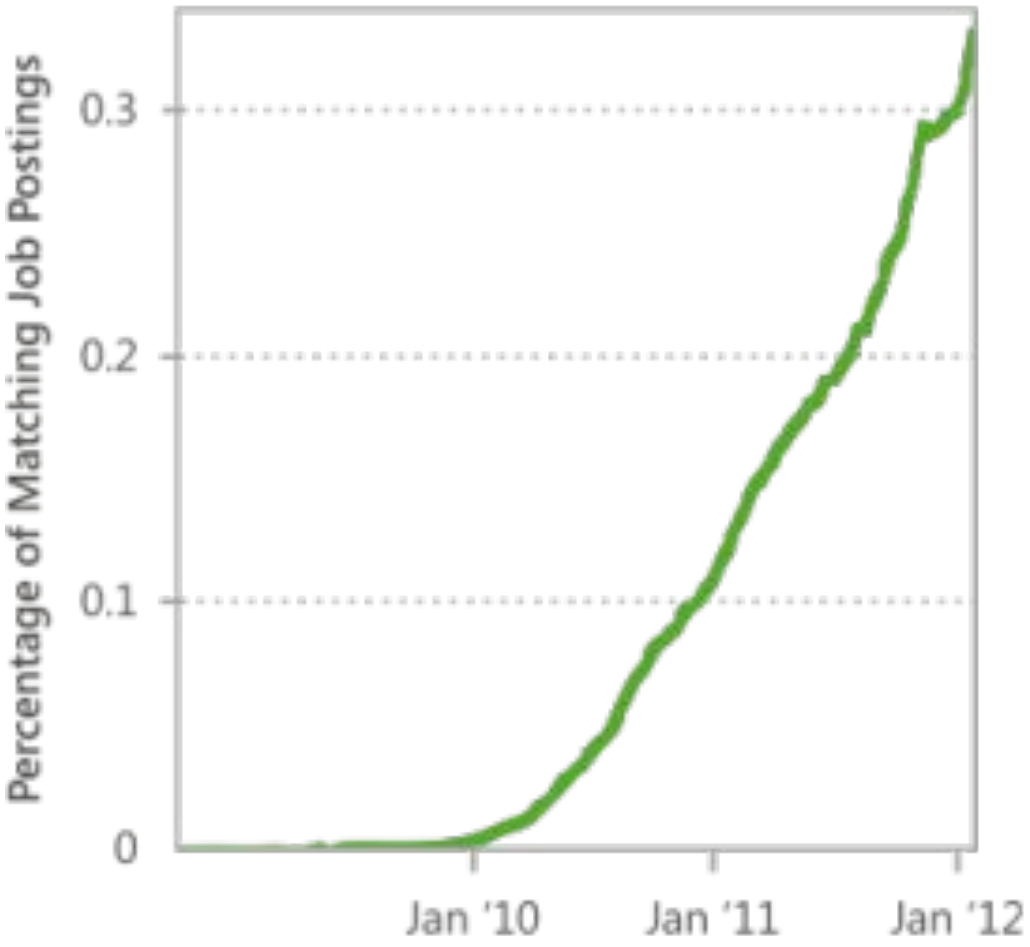
Une compétence en demande sur LinkedIn



Une adoption rapide selon Google search



2ème trend dans les jobs de Indeed.com



Top Job Trends

1. HTML 5
2. **MongoDB**
3. iOS
4. Android
5. Mobile Apps
6. Puppet
7. Hadoop
8. jQuery
9. PaaS
10. Social Media

20,000,000

téléchargements

8% x 40 milliards $

That’s a lot of money

2000 utilisateurs payants

And a lot of users

33% cies fortune 500

Total success!

Installer MongoDB

https://www.mongodb.com/download-center (onglet server)

On installe la base de données avec les options par défaut

- dans C:\MongoDB
- on crée C:\data\db + permissions

Pour tester dans 2 consoles (serveur/client)

- C:\MongoDB\Server\4.0\bin>mongod
- C:\MongoDB\Server\4.0\bin>mongo

Commandes du client mongo

C:\MongoDB\Server\5.0\bin>mongo

- MongoDB Enterprise > help
- MongoDB Enterprise > show dbs

Après la programmation

- MongoDB Enterprise > use **bergerie**
- MongoDB Enterprise > show collections
- MongoDB Enterprise > db.**mouton**.find()

> help

```
db.help()                    help on db methods
db.mycoll.help()             help on collection methods
sh.help()                    sharding helpers
rs.help()                    replica set helpers
help admin                   administrative help
help connect                 connecting to a db help
help keys                    key shortcuts
help misc                    misc things to know
help mr                      mapreduce
show dbs                     show database names
show collections             show collections in current database
show users                   show users in current database
show profile                 show most recent system.profile entries with time >= 1ms
show logs                    show the accessible logger names
show log [name]              prints out the last segment of log in memory, 'global' is default
use <db_name>                set current database
db.foo.find()                list objects in collection foo
db.foo.find( { a : 1 } )     list objects in foo where a == 1
it                           result of the last line evaluated; use to further iterate
DBQuery.shellBatchSize = x   set default number of items to display on shell
exit                         quit the mongo shell
```

Télécharger le Driver

Soit on peut composer le tout soi-même

- mongodb-driver
- bson
- core-driver

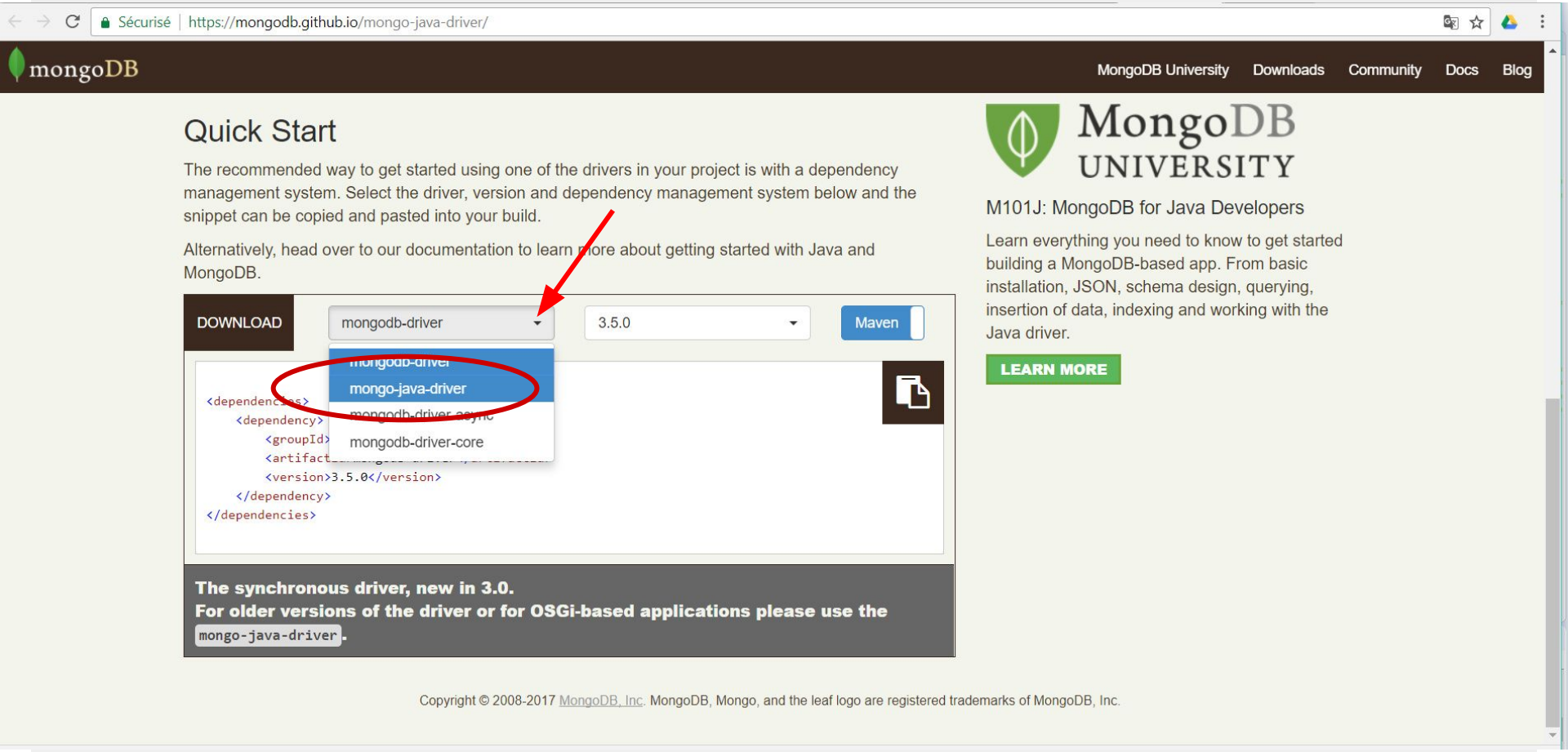
https://mongodb.github.io/mongo-java-driver/

Soit on peut utiliser le paquet pré-assemblé !

- mongo-java-driver

https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver/3.0.0

Télécharger le Driver



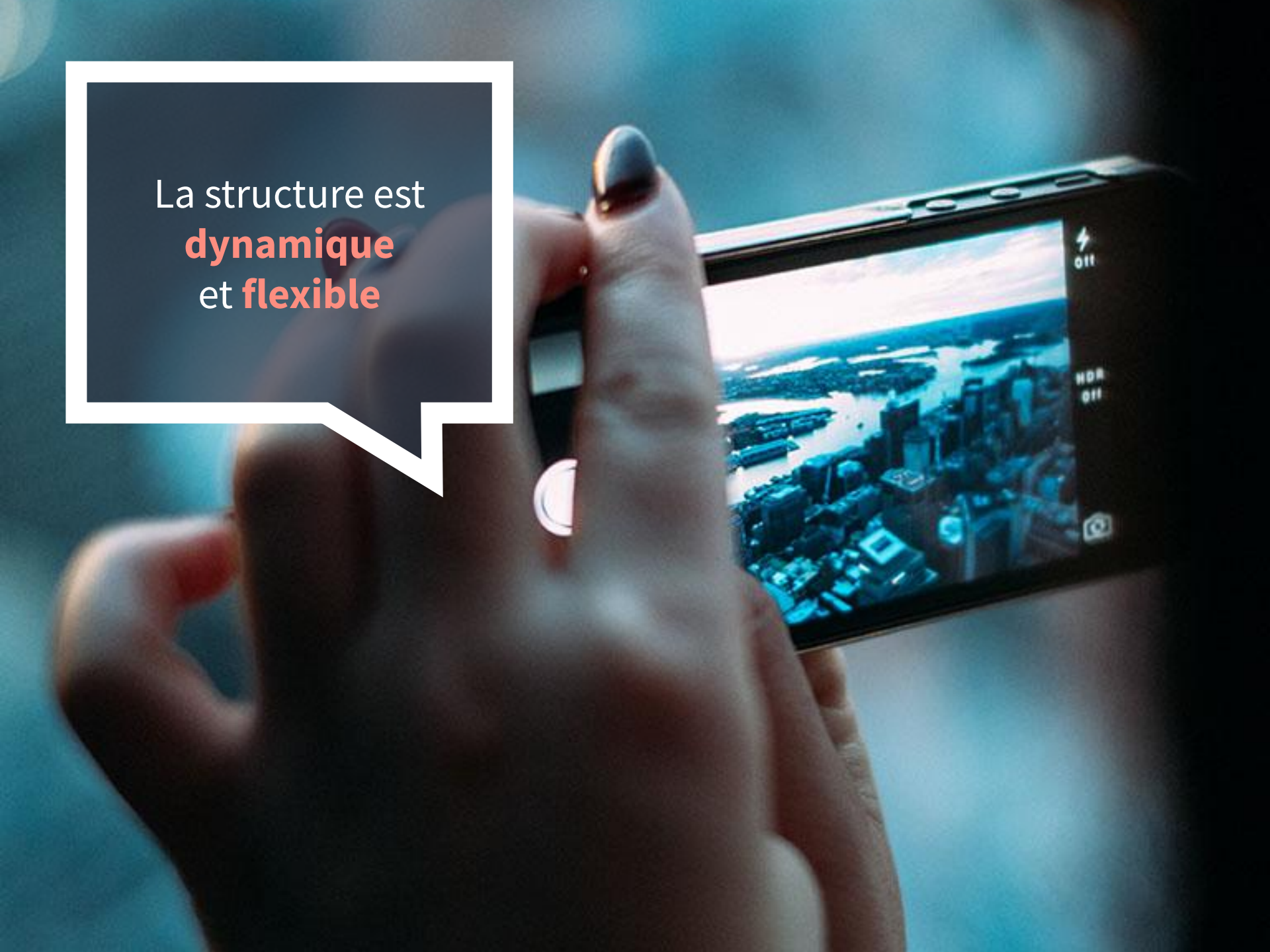
Le projet test

Il faut inclure la bibliothèque

- On place lib/mongo-java-driver-3.5.0.jar
- On l'ajoute sur le path (relatif)

On programme avec

- import com.mongodb.MongoClient;
- import com.mongodb.DB;
- import com.mongodb.DBCollection;
- import com.mongodb.DBCursor;
- import com.mongodb.DBObject;


La structure est
dynamique
et flexible

Courriel de bienvenue

Hi Nadine,

Now that you have installed your trial of MongoDB Enterprise, it is time to familiarize yourself with CRUD operations.

Here are some docs I think you will find helpful:

- MongoDB CRUD Operations Overview
- Insert Documents
- Query Documents
- Delete Documents
- Update Documents

Let me know if you need anything.

- Andrew

PS - Check out the Getting Started with MongoDB Hub for our most helpful videos, tutorials, and articles

La connection

```
@SuppressWarnings("deprecation")

// Me connecter (et créer) la base de données
MongoClient mongoClient = new MongoClient();
DB basededonnees = mongoClient.getDB("bergerie");

// Fermer la base de données et la connection
mongoClient.close();
```

Si la base de données n'existe pas, elle sera créée dynamiquement. Pas besoin de la créer en avance.

Ajouter un objet

```
// Ajouter un mouton
DBCollection listeMoutons = basededonnees.getCollection("mouton");

DBObject moutonMongo = new BasicDBObject();
        Map moutonMap = new HashMap();
        moutonMap.put("nom", "Billy");
        moutonMap.put("description", "Matinal");
        moutonMap.put("couleur", "Roux");
moutonMongo.putAll(moutonMap);

listeMoutons.insert(moutonMongo);
```

Ajouter un objet

```
// Ajouter un mouton
Map moutonMap = new HashMap();
moutonMap.put("nom", "Billy");
moutonMap.put("description", "Matinal");
moutonMap.put("couleur", "Roux");

DBObject moutonMongo = new BasicDBObject();
moutonMongo.putAll(moutonMap);

DBCollection listeMoutons = basededonnees.getCollection("mouton");
listeMoutons.insert(moutonMongo);
```

Récupérer un objet

```
// Trouver un mouton
DBObject critereMouton = new BasicDBObject("nom","Billy");
DBCursor pointeurMouton = listeMoutons.find(critereMouton);
moutonMap = pointeurMouton.one().toMap();
System.out.println(moutonMap.get("description"));
```

Codes d'accès pour CRUD - deprecated actuel

MongoDB	Code JAVA http://mongodb.github.io/mongo-java-driver/3.6/ http://mongodb.github.io/mongo-java-driver/3.6/driver/	Shell + JavaScript https://docs.mongodb.com/v3.4/reference/method/js-collection/
Trouver	**uneCollection.find(DBObject)**	
Ajouter	**collection.insert(HashMap)** **collection.insertOne(Document)** **collection.insertMany(List<Document>)**	
Modifier	**collection.updateOne(eq("id", 10), new Document("$set", new Document("resume","joyeux")));**	**db.mouton.update({ id: "5" }, {resume: "joyeux" }); (voir doc)**
Effacer	**collection.remove(objetCritere);**	

Codes d'accès pour CRUD - vieux code

MongoDB	Code JAVA http://mongodb.github.io/mongo-java-driver/3.6/ http://mongodb.github.io/mongo-java-driver/3.6/driver/	Shell + JavaScript https://docs.mongodb.com/v3.4/reference/method/js-collection/
Trouver	**uneCollection.find(DBObject)**	
Ajouter	**collection.insert(HashMap)** **collection.insertOne(Document)** **collection.insertMany(List<Document>)**	
Modifier	**collection.updateOne(eq("id", 10), new Document("$set", new Document("resume","joyeux")));**	**db.mouton.update({ id: "5" }, {resume: "joyeux" }); (voir doc)**
Effacer	**collection.deleteOne(eq("id", 110));**	

Code total ajouter

```
MongoClient mongo = new MongoClient();
DB basededonnees = mongo.getDB("bergerie");

Map valeursMouton = new HashMap();
valeursMouton.put("nom","Lion");
valeursMouton.put("couleur","verte");

DBObject mouton = new BasicDBObject();
mouton.putAll(valeursMouton);

DBCollection listeMoutons = basededonnees.getCollection("mouton");
listeMoutons.insert(mouton);

mongo.close();
```

Lien avec l'architecture

Nous avons utilisé un type Map plutôt qu'un type DBObject pour initialiser le mouton, et ce afin de s'intégrer plus facilement avec le modèle MVC existant.

La meilleure manière de produire les modèle en Map N'EST PAS de les fabriquer dans les contrôleurs ou DAO, mais de les demander aux modèles sémantiques des objets et de les programmer spécifiquement dans chaque modèle (ou de les faire générer par une classe de base en framework).

En résumé, on fera la méthode Mouton.convertirEnHash( )

Éviter la déconnection

2 conseils dans
https://stackoverflow.com/questions/35597930/sockettimeout-with-opened-connection-in-mongodb/35777711

1)

```
MongoCredential acces = MongoCredential.createCredential("nadine", "bergerie", "motdepasse".toCharArray());
String uri = "mongodb://nadine:motdepasse@mon-url.com:27017";
String serveur = "mon-url.com:27017";
MongoClientOptions.Builder options = MongoClientOptions.builder();
options.socketKeepAlive(true);
MongoClient mongo = new MongoClient(new ServerAddress(serveur), Arrays.asList(acces), options.build());
```

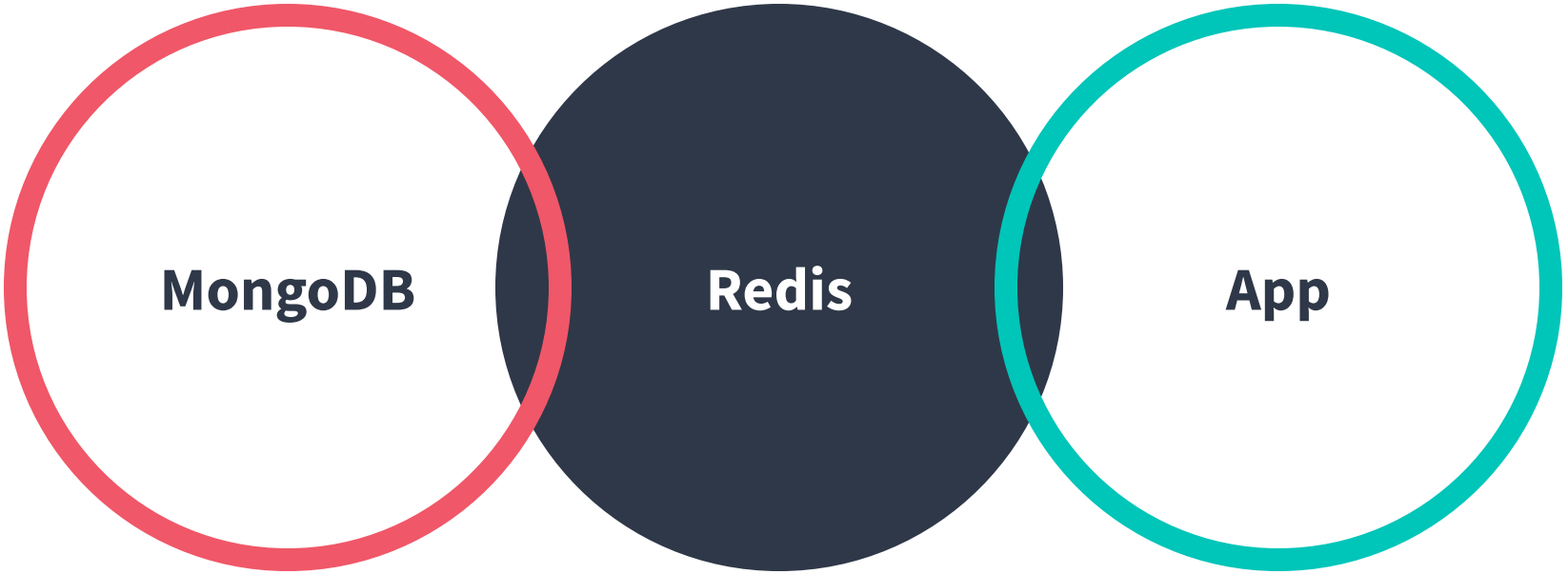
2) sudo sysctl -w net.ipv4.tcp_keepalive_time=120

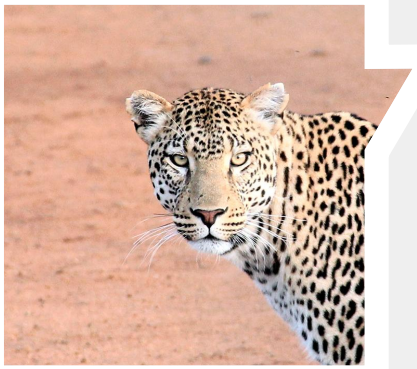


"True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information."

WINSTON CHURCHILL

Architecture multi-système possible





Bravo!

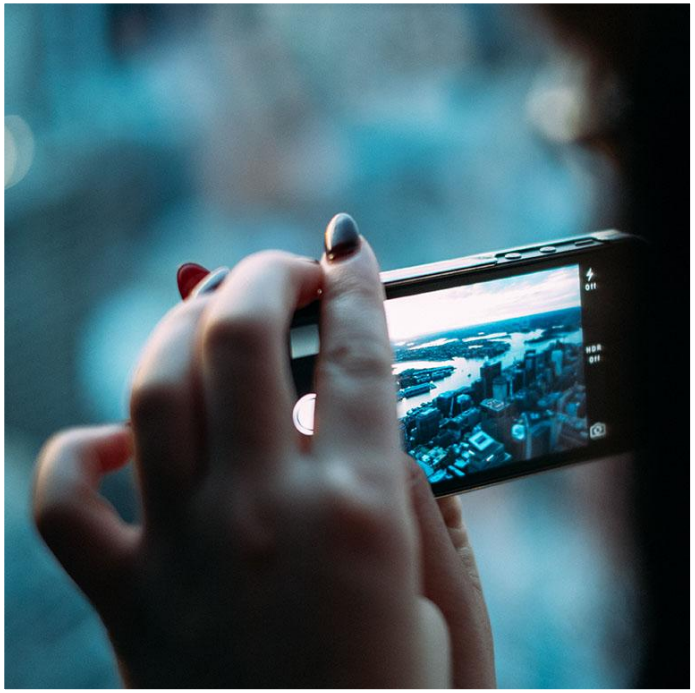
Vous avez programmé du NoSQL

Si vous avez des questions, vous pouvez me rejoindre à

nadineducegep@gmail.com

...

...



Instructions for use

Open this document in Google Slides (if you are at slidescarnival.com use the button below this presentation)

You have to be signed in to your Google account

EDIT IN GOOGLE SLIDES

Go to the ***File*** menu and select ***Make a copy***.

You will get a copy of this document on your Google Drive and will be able to edit, add or delete slides.

EDIT IN POWERPOINT®

Go to the ***File*** menu and select ***Download as Microsoft PowerPoint***. You will get a .pptx file that you can edit in PowerPoint.

Remember to download and install the fonts used in this presentation (you'll find the links to the font files needed in the Presentation design slide)

More info on how to use this template at www.slidescarnival.com/help-use-presentation-template

This template is free to use under Creative Commons Attribution license. You can keep the Credits slide or mention SlidesCarnival and other resources used in a slide footer.

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

Presentation design

This presentations uses the following typographies and colors:

- Titles & body copy: **Source Sans Pro**

You can download the fonts on this page:

https://www.google.com/fonts#UsePlace:use/Collection:Source+Sans+Pro:400,700,400italic,700italic

Click on the “arrow button” that appears on the top right 

- Dark gray **#2f3848**
- Turquoise **#00c5b9**
- Salmon **#f05768**

You don’t need to keep this slide in your presentation. It’s only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®

Mongo + Node

Particularites non-documentees

MongoDB avec node

pour avoir id en tant qu'id lisible - faut avoir fait un toString()

quand aggregat sur collection - faire toArray pour que aggregat s'execute() - a cause du lazy loading;

TEMPLATE

TEMPLATE

TEMPLATE

You can also split your content

White

Is the color of milk and fresh snow, the color produced by the combination of all the colors of the visible spectrum.

Black

Is the color of coal, ebony, and of outer space. It is the darkest color, the result of the absence of or complete absorption of light.

In two or three columns

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

Let’s review some concepts



Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.



Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.



Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.



Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.



Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

Utilisation du nouveau module dans NodeJS (Electron)

```
npm install mongodb --save
```

API

https://mongodb.github.io/node-mongodb-native/